Wikipedia
The Free Encyclopedia

Universally unique identifier

A **universally unique identifier** (**UUID**) is a 128-bit label used for information in computer systems. The term **globally unique identifier** (**GUID**) is also used.[1]

When generated according to the standard methods, UUIDs are, for practical purposes, unique. Their uniqueness does not depend on a central registration authority or coordination between the parties generating them, unlike most other numbering schemes. While the probability that a UUID will be duplicated is not zero, it is generally considered close enough to zero to be negligible.[2][3]

Thus, anyone can create a UUID and use it to identify something with near certainty that the identifier does not duplicate one that has already been, or will be, created to identify something else. Information labeled with UUIDs by independent parties can therefore be later combined into a single database or transmitted on the same channel, with a negligible probability of duplication.

Adoption of UUIDs is widespread, with many computing platforms providing support for generating them and for parsing their textual representation.

Universally unique identifier

UUID/GUID as used by UEFI variables

Acronym	UUID
Organisation	Open Software Foundation (OSF) ISO/IEC Internet Engineering Task Force (IETF)
No. of digits	32
Example	`123e4567-e89b-12d3-a456-426614174000`
Website	RFC 4122 (https://datatracker.ietf.org/doc/html/rfc4122)

History

In the 1980s Apollo Computer originally used UUIDs in the Network Computing System (NCS) and later in the Open Software Foundation's (OSF) Distributed Computing Environment (DCE). The initial design of DCE UUIDs was based on the NCS UUIDs,[4] whose design was in turn inspired by the (64-bit) unique identifiers defined and used pervasively in Domain/OS, an operating system designed by Apollo Computer. Later, the Microsoft Windows platforms adopted the DCE design as "globally unique identifiers" (GUIDs). RFC 4122 registered a URN namespace for UUIDs[1] and recapitulated the earlier specifications, with the same technical content. When in July 2005 RFC 4122 was published as a proposed IETF standard, the ITU had also standardized UUIDs, based on the previous standards and early versions of RFC 4122.

Standards

UUIDs are standardized by the Open Software Foundation (OSF) as part of the Distributed Computing Environment (DCE).[5][6]

UUIDs are documented as part of ISO/IEC 11578:1996 "Information technology – Open Systems Interconnection – Remote Procedure Call (RPC)" and more recently in ITU-T Rec. X.667 | ISO/IEC 9834-8:2005.[7]

The Internet Engineering Task Force (IETF) published the Standards-Track RFC 4122,[1] technically equivalent to ITU-T Rec. X.667 | ISO/IEC 9834-8.

Format

In its canonical textual representation, the 16 octets of a UUID are represented as 32 hexadecimal (base-16) digits, displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters (32 hexadecimal characters and 4 hyphens). For example:

```
123e4567-e89b-12d3-a456-426614174000
xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxx
```

The four-bit *M* and the 1- to 3-bit *N* fields code the format of the UUID itself.

The four bits of digit `M` are the UUID version, and the 1 to 3 most significant bits of digit `N` code the UUID variant. (See below.) In the example, *M* is `1`, and *N* is `a` ($10xx_2$), meaning that this is a version-1, variant-1 UUID; that is, a time-based DCE/RFC 4122 UUID.

The canonical 8-4-4-4-12 format string is based on the record layout for the 16 bytes of the UUID:[1]

UUID record layout

Name	Length (bytes)	Length (hex digits)	Length (bits)	Contents
time_low	4	8	32	integer giving the low 32 bits of the time
time_mid	2	4	16	integer giving the middle 16 bits of the time
time_hi_and_version	2	4	16	4-bit "version" in the most significant bits, followed by the high 12 bits of the time
clock_seq_hi_and_res clock_seq_low	2	4	16	1 to 3-bit "variant" in the most significant bits, followed by the 13 to 15-bit clock sequence
node	6	12	48	the 48-bit node id

These fields correspond to those in version 1 and 2 UUIDs (that is, time-based UUIDs), but the same 8-4-4-4-12 representation is used for all UUIDs, even for UUIDs constructed differently.

RFC 4122 Section 3 (https://tools.ietf.org/html/rfc4122#section-3) requires that the characters be generated in lower case, while being case-insensitive on input.

Microsoft GUIDs are sometimes represented with surrounding braces:

```
{123e4567-e89b-12d3-a456-426652340000}
```

This format should not be confused with "Windows Registry format", which refers to the format *within* the curly braces.[8]

RFC 4122 defines a Uniform Resource Name (URN) namespace for UUIDs. A UUID presented as a URN appears as follows:[1]

```
urn:uuid:123e4567-e89b-12d3-a456-426655440000
```

Encoding

The binary encoding of UUIDs varies between systems. Variant 1 UUIDs, nowadays the most common variant, are encoded in a big-endian format. For example, `00112233-4455-6677-8899-aabbccddeeff` is encoded as the bytes `00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff`.[9][10]

Variant 2 UUIDs, historically used in Microsoft's COM/OLE libraries, use a little-endian format, but appear mixed-endian with the first three components of the UUID as little-endian and last two big-endian, due to the missing byte dashes when formatted as a string.[11] For example, `00112233-4455-6677-c899-aabbccddeeff` is encoded as the bytes `33 22 11 00 55 44 77 66 88 99 aa bb cc dd ee ff`.[12][13] See the section on Variants for details on why the '88' byte becomes 'c8' in Variant 2.

Variants

The "variant" field of UUIDs, or the *N* position indicate their format and encoding. RFC 4122 defines four variants of lengths 1 to 3 bits:

- Variant 0 (indicated by the one-bit pattern $0xxx_2$, *N* = `0..7`) is for backwards compatibility with the now-obsolete Apollo Network Computing System 1.5 UUID format developed around 1988. The first 6 octets of the UUID are a 48-bit timestamp (the number of 4-microsecond units of time since 1 January 1980 UTC); the next 2 octets are reserved; the next octet is the "address family"; and the final 7 octets are a 56-bit host ID in the form specified by the address family. Though different in detail, the similarity with modern version-1 UUIDs is evident. The variant bits in the current UUID specification coincide with the high bits of the address family octet in NCS UUIDs. Though the address family could hold values in the range 0..255, only the values 0..13 were ever defined. Accordingly, the variant-0 bit pattern `0xxx` avoids conflicts with historical NCS UUIDs, should any still exist in databases.[14]
- Variant 1 ($10xx_2$, *N* = `8..b`, 2 bits) are referred to as RFC 4122/DCE 1.1 UUIDs, or "Leach–Salz" UUIDs, after the authors of the original Internet Draft.
- Variant 2 ($110x_2$, *N* = `c..d`, 3 bits) is characterized in the RFC as "reserved, Microsoft Corporation backward compatibility" and was used for early GUIDs on the Microsoft Windows platform. It differs from variant 1 only by the endianness in binary storage or transmission: variant-1 UUIDs use "network" (big-endian) byte order, while variant-2 GUIDs use "native" (little-endian) byte order for some subfields of the UUID.
- Reserved is defined as the 3-bit variant bit pattern $111x_2$ (*N* = `e..f`).

Variants 1 and 2 are used by the current UUID specification. In their textual representations, variants 1 and 2 are the same, except for the variant bits. In the binary representation, there is an endianness difference.[1] When byte swapping is required to convert between the big-endian byte order of variant 1 and the little-endian byte order of variant 2, the fields above define the swapping. The first three fields are unsigned 32- and 16-bit integers and are subject to swapping, while the last two fields consist of uninterpreted bytes, not subject to swapping. This byte swapping applies even for versions 3, 4, and 5, where the canonical fields do not correspond to the content of the UUID.[1]

While some important GUIDs, such as the identifier for the Component Object Model IUnknown interface, are nominally variant-2 UUIDs, many identifiers generated and used in Microsoft Windows software and referred to as "GUIDs" are standard variant-1 RFC 4122/DCE 1.1 network-byte-order UUIDs, rather than little-endian variant-2 UUIDs. The current version of the Microsoft `guidgen` tool produces standard variant-1 UUIDs. Some Microsoft documentation states that "GUID" is a synonym for "UUID",[15] as standardized in RFC 4122. RFC 4122 itself states that UUIDs "are also known as GUIDs". All this suggests that "GUID", while originally referring to a variant of UUID used by Microsoft, has become simply an alternative name for UUID, with both variant-1 and variant-2 GUIDs being extant.